



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/021,535	12/12/2001	Jeffrey John Anuszczyk	111345.122	1826
7590	03/08/2007		EXAMINER SHAW, PEILING ANDY	
Joseph F. Haag Hale and Dorr LLP 60 State Street Boston, MA 02106			ART UNIT 2144	PAPER NUMBER
SHORTENED STATUTORY PERIOD OF RESPONSE 3 MONTHS		MAIL DATE 03/08/2007	DELIVERY MODE PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)
	10/021,535	ANUSZCZYK ET AL.
	Examiner Peling A. Shaw	Art Unit 2144

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 04 December 2006.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 2-30 and 32-40 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 2-30 and 32-40 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

1. Amendment received on 12/04/2006 has been entered into record. No claim is amended.

Claims 2-30 and 32-40 are currently pending.

2. Applicant's submission filed on 03/15/2006 was entered. Claims 2, 4-5, 11-12, 14-15, 30 and 40 were amended. Claims 1, 31 and 41-44 were cancelled.

3. Amendment received on 06/29/2005 was entered. Claims 1, 5, 14-15, 30, 40-41 and 43-44 were amended. Claims 1-44 are still pending.

Priority

4. This application has no priority claim made. The filing date is 12/12/2001.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 30 and 32-40 are rejected under 35 U.S.C. 102(a) as being anticipated by Keller et al. (Dynamic Dependencies in Application Service Management), hereinafter referred as Keller.

a. Regarding claim 30, Keller disclosed a method for determining dependencies between at least two components in an information technology (IT) system, comprising: discovering the at least two components in the IT system by receiving real-time messages (page 5, right column, last paragraph: dependencies identified at application installation time and discovered at runtime, functional dependencies between application and service categories, the structural part captures dynamic

information related to concrete service implementation; page 6, right column, 1st and 2nd paragraphs: MLMs, event reception and forwarding, resource discovery functions, application operating on top of a network management platform, e.g. DNS, web hosting and firewall); monitoring the usage of resources by the two components in the IT system and, if a resource is used by one of the two components, generating a message indicating the use of that resource by that component (page 4, left column, 1st paragraph: compute the runtime dependencies of the Name Service, Detailed configuration parameters of a WWW server and allow the forecast the runtime behavior of a web server; page 5, right column, last paragraph: dependencies identified at application installation time and discovered at runtime; page 6, right column, 1st paragraph: MLMs, event reception and forwarding, resource discovery functions, collected data describes, for each end-to-end application services, the dependencies it has on lower level application and network layer services and components); accumulating each message indicating the use of one of the resources by one of the two components (page 5, right column, last paragraph-page 6, left column, 1st paragraph: event reception and forwarding, dependencies identified at application install time and those discovered at runtime, structural part captures dynamic information; page 6, right column, 3rd paragraph-page 7, left column, 1st paragraph: application service agent); and if the accumulated messages indicate that the two components use the same resource, then indicating that a dependency between the two components has been detected (page 5, right column, 4th column; page 6, right column, 3rd paragraph-page 7, left column, 1st paragraph: application

service agent), and determining a direction of the dependency between the two components (page 2, left column, 2nd paragraph, item 1).

- b. Regarding claim 32, Keller disclosed the method of claim 30, wherein the component is selected from the group consisting of an application, a network connection endpoint, and a server (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 6, right column, 1st paragraph).
- c. Regarding claim 33, Keller disclosed the method of claim 32, wherein at least one message indicates a network outbound connection by one of the two components (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 6, right column, 1st paragraph).
- d. Regarding claim 34, Keller disclosed the method of claim 32, wherein at least one message indicates a network listener by one of the two components (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 6, right column, 1st paragraph).
- e. Regarding claim 35, Keller disclosed the method of claim 32, wherein at least one message indicates a use of a file by one of the two components (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 3, right column: Component Type and Component Activity).
- f. Regarding claim 36, Keller disclosed the method of claim 30, further comprising tracking changes to the dependency between the two components (page 6, right column, 1st paragraph-page 7, left column, 6th paragraph: MLMs, Application Service Agent, Resource Broker and Resource Directory).

- g. Regarding claim 37, Keller disclosed the method of claim 30, wherein the dependency is a containment dependency (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 3, right column: Component Type and Component Activity: file and file system).
- h. Regarding claim 38, Keller disclosed the method of claim 30, wherein the dependency is a network dependency (Fig. 1; page 2, left column, 1st paragraph and 2nd paragraph, item 1; page 6, right column, 1st paragraph).
- i. Regarding claim 39, Keller disclosed the method of claim 30, wherein the dependency is a shared usage dependency (Fig. 1, IP Provider and Network Provider; page 2, left column, 1st paragraph and 2nd paragraph, item 1).
- j. Regarding claim 40, Keller disclosed an apparatus for determining dependencies between at least two components in an information technology (IT) system, comprising: A memory storing a program (page 6, right column, 2nd paragraph: application service agent); A processor in communication with the memory (page 6, right column, 2nd paragraph: application service agent); in which the processor is directed by the program to: discover the at least two components in the IT system (page 6, right column, 1st paragraph-page 7, left column, 6th paragraph: application service agent work through MLM, Resource Broker and Resource Directory); monitor the usage of resources by the two components in the IT system by receiving real-time messages and, if a resource is used by one of the two components, generating a message indicating the use of that resource by that component (page 6, right column, 1st paragraph-page 7, left column, 1st paragraph: application service

agent work through MLM); accumulate each message indicating the use of one of the resources by one of the two components (page 6, right column, 3rd paragraph-page 7, left column, 1st paragraph: application service agent); and if the accumulated messages indicate that the two components use the same resource, then indicate that a dependency between the two components has been detected (page 5, right column, 4th column; page 6, right column, 3rd paragraph-page 7, left column, 1st paragraph: application service agent); and determining a direction of the dependency between the two components (page 2, left column, 2nd paragraph, item 1).

Keller disclosed all limitations of claims 30 and 32-40. Claims 30 and 32-40 are rejected under 35 U.S.C. 102(a).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 2-12 and 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kar et al., (An Architecture for Managing Application Services over Global Networks), in view of Kathrow, et al. (US 6393438 B1), hereinafter referred as Kathrow.

a. Kar shows (claim 5) a method for collecting information on components in an information technology (IT) system, comprising: discovering the existence of at least one of the components in the IT system (page 2, right column, 2nd paragraph- page 3,

left column, 1st paragraph: MLM, resource discovery functions); determining at least one dependency between two or more of the components (page 3, right column, 3rd-5th paragraphs: Dependency Analysis for Service Management); and tracking changes to at least one of the components and the dependency between two or more of the components (page 2, right column, 2nd paragraph- page 3, left column, 1st paragraph: MLM; page 5, right column, 2nd paragraph-page 6, left column, 4th paragraph: Application Service Agents, Resource Broker and Resource Directory). Kar does not show explicitly (claim 5) discovering the existence of at least one of the components by receiving real-time messages and using at least one fingerprint, by receiving the real-time messages as event information regarding an occurrence in the IT system, the occurrence relating to a first component; comparing the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component; and determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched. However, Kar does show using resource identifier (page 6, right column, item 1) and software definition file format with many attributes (page 7, left column, 2nd paragraph).

- b. Kathrow shows (claim 5) discovering the existence of at least one of the components by receiving real-time messages and using at least one fingerprint, by receiving the real-time messages as event information regarding an occurrence in the IT system, the occurrence relating to a first component (Fig. 2, item 252; column 4, line 7-18); comparing the first component along with other components to at least one

fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component (Fig. 4, item 432; column 11, line 49-59); and determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched (Fig. 4, items 432, 434 and 440; column 11, line 59-62) in an analogous art for the purpose of identifying the existence of differences between two files based upon the fingerprints.

- c. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Kar's Mid-level manger to include Kathrow's identifying the difference of two files based upon the fingerprint, including device descriptors.
- d. The modification would have been obvious because one of ordinary skill in the art would have been motivated to incorporate the file/device record updates base upon the hash technologies, i.e. fingerprint/digital signature, to improve the security and performance of configuration management per Kathrow's teaching in the management of applications per Kar (page 2, left column, 2nd paragraph- page 3, left column, 1st paragraph: using mid level manager to monitor the health and status of a application service) and Kathrow (column 1, lines 36-52: using window registry to manage computer programs running under window operating system) 's teachings.
- e. Regarding claim 2, Kar shows further comprising generating a visual map of the IT system, the visual map including a depiction of at least one of the components and the at least one dependency between two or more of the components (Fig. 4; page 4, left

column, 2nd paragraph: Dependency Graph; page 5, left column, last paragraph, and right column, last paragraph-page 6, left column, 1st paragraph).

- f. Regarding claim 3, Kar shows wherein the visual map includes tracked changes to at least one of the components (page 5, right column, last paragraph-page 6, left column, 1st paragraph).
- g. Regarding claim 4, Kar shows wherein at least one of the components is an application (page 5, right column, 2nd paragraph-page 6, left column, 1st paragraph: Application Service Agents).
- h. Regarding claim 6, Kathrow shows wherein the occurrence is selected from one or more of a file creation, a file deletion, and a file modification (column 4, line 60-65).
- i. Regarding claim 7, Kathrow shows wherein the occurrence is selected from one or more of a registry key creation, a registry key deletion, and a registry key modification (column 4, line 60-65).
- j. Regarding claim 8, Kathrow shows wherein the occurrence is information regarding detection of a particular component in the IT system (column 4, line 60-65).
- k. Regarding claim 9, Kathrow shows further comprising indicating that a particular component has been damaged if the occurrence is a deletion and at least one of the elements of the fingerprint are no longer matched by the components in the IT system (Fig. 4, item 436; column 4, line 60-65; column 11, line 62-64; column 14, line 8-15).
- l. Regarding claim 10, Kathrow shows further comprising indicating that a particular component has been uninstalled if the occurrence is a deletion and all of the elements of a minimum set of the fingerprint are no longer matched by the components in the

IT system (Fig. 4, item 436; column 4, line 60-65; column 11, line 62-64; column 14, line 8-15).

- m. Regarding claim 11, Kar shows wherein the at least one dependency is selected from the group consisting of shared library usage, network usage, and containment dependencies (page 3, right column, 4th-last paragraphs: DNS, NFS, IP service, PVC, network components, servers and applications).
- n. Regarding claim 12, Kar shows further comprising: generating a component discovered message upon the discovery of one of the components; retrieving a list of elements to track for the discovered component; and using the list of elements to track changes to the discovered component (page 2, right column, 2nd paragraph, 1st-3rd bullets).
- o. Claims 14-15 are of the same scope as claim 5. These are rejected for the same reasons as for claim 5.

Together Kar and Kathrow disclosed all limitations of claims 2-12 and 14-15. Claims 2-12 and 14-15 are rejected under 35 U.S.C. 103(a).

7. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kar et al., (An Architecture for Managing Application Services over Global Networks), hereinafter referred as Kar, in view of Kathrow, et al. (US 6393438 B1), hereinafter referred as Kathrow.

- a. Kar shows an agent for collecting information on components in an information technology (IT) system, the agent residing on a computer in the IT system, the agent comprising: an observer module to detect event information about elements of the computer (page 2, right column, 2nd paragraph, 2nd bullet: MLM); and an analysis

module to process the event information, the analysis module including: (A) component discovery rules to process event information and using an accumulator to discover the existence on the IT system of at least one of the components (page 2, right column, 2nd paragraph, 2nd bullet: MLM; page 5, right column, 2nd paragraph-page 6, left column, 4th paragraph: Application Service Agents, Resource Broker and Resource Directory), and (B) dependency discovery rules to detect relationships between components of the IT system (page 5, left column, last paragraph). Kar does not show matching event information with elements of one or more fingerprints of known components. However, Kar does show using resource identifier (page 6, right column, item 1) and software definition file format with many attributes (page 7, left column, 2nd paragraph).

- b. Kathrow shows matching event information with elements of one or more fingerprints of known components (Fig. 4, items 432, 434 and 440; column 11, line 49-62) in an analogous art for the purpose of identifying the existence of differences between two files based upon the fingerprints.
- c. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Kar's Mid-level manager to include Kathrow's identifying the difference of two files based upon the fingerprint, including device descriptors.
- d. The modification would have been obvious because one of ordinary skill in the art would have been motivated to incorporate the file/device record updates base upon the hash technologies, i.e. fingerprint/digital signature, to improve the security and

performance of configuration management per Kathrow's teaching in the management of applications per Kar (page 2, left column, 2nd paragraph- page 3, left column, 1st paragraph: using mid level manager to monitor the health and status of a application service) and Kathrow (column 1, lines 36-52: using window registry to manage computer programs running under window operating system) 's teachings.

Together Kar and Kathrow disclosed all limitations of claim 13. Claim 13 is rejected under 35 U.S.C. 103(a).

8. Claims 16-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kathrow, et al. (US 6393438 B1), hereinafter referred as Kathrow, in view of Kar et al., (An Architecture for Managing Application Services over Global Networks), hereinafter referred as Kar.

a. Regarding claims 16, 23 and 24, Kathrow shows a method, a computer-readable medium storing instructions that direct a microprocessor, and an apparatus with memory containing a program and a processor executing the program for discovering components in an information technology (IT) system, comprising: receiving event information regarding an occurrence in the IT system, the occurrence relating to a first component (Fig. 2, item 252; column 4, line 7-18); comparing the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component (Fig. 4, items 432, 434 and 440; column 11, line 49-62). Kathrow also shows (column 4, 1st paragraph; column 11, line 5-64) a single file could be divided into multiple files and multiple fingerprints are used. Kathrow does not show if the first component and the other discovered components match substantially all of the key low-level

elements of the fingerprint, using A subfingerprint of A known refined component to discover the existence of A second component that corresponds to the known refined component. However, Kar does show using resource identifier (page 6, right column, item 1) and software definition file format with many attributes (page 7, left column, 2nd paragraph).

- b. Kar shows (page 5, left column, last paragraph, and right column, last paragraph-page 6, left column, 1st paragraph) as a result of static analysis, the resource dependency graph could be constructed in an analogous art for the purpose of determining the resource dependency in the application service management.
- c. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Kathrow's file comparison based upon the fingerprint method to include a further depending file (refined component) comparison through an application control (Fig. 2, item 252; column 4, line 16-18) via the result of fingerprint comparison (Fig. 2, item 254; Fig. 4, items 432, 434 and 440; column 11, line 59-64) to further ensure all the dependent files as suggested (column 4, 1st paragraph; column 11, line 62-64) will be compared based upon fingerprints.
- d. The modification would have been obvious because one of ordinary skill in the art would have been motivated to fingerprint check not only one file out of multiple files that were from the division out of a single file, but multiple files to the integrity of a file (all divided files or associated resources) in the management of applications per Kar (page 2, left column, 2nd paragraph- page 3, left column, 1st paragraph: using mid level manager to monitor the health and status of a application service) and

Kathrow (column 1, lines 36-52: using window registry to manage computer programs running under window operating system) 's teachings.

- e. Regarding claim 17, Kathrow shows the method of claim 16, wherein the known refined component is a version of the known component (column 10, line 16-20).
- f. Regarding claim 18, Kathrow shows the method of claim 16, wherein the known refined component is an optional piece of the known component (column 12, line 6-21).
- g. Regarding claim 19, Kathrow shows the method of claim 16, further comprising generating a command message to collect further information if all of the low-level elements of the fingerprint are matched (column 11, line 59-62).
- h. Regarding claim 20, Kathrow shows the method of claim 19, further comprising receiving event information in response to the command message, wherein the event information is used with the subfingerprint of the known refined component to discover the existence of the second component (Fig. 2, item 252 and 254; column 4, line 7-18).
- i. Regarding claim 21, Kathrow shows the method of claim 16, further comprising detecting low-level items in the IT systems and generating event information regarding the low-level items (Kar, page 5, left column, last paragraph, and right column, last paragraph-page 6, left column, 1st paragraph; Kathrow, Fig. 2, item 252; column 4, line 7-18).

j. Regarding claim 22, Kathrow shows the method of claim 21, wherein the low-level items are selected from one or more of files, registry settings, and database schemas (column 3, line 59-62).

Together Kathrow and Kar disclosed all limitations of claims 16-24. Claims 16-24 are rejected under 35 U.S.C. 103(a).

9. Claims 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kathrow, et al. (US 6393438 B1), hereinafter referred as Kathrow, in view of Kar et al., (An Architecture for Managing Application Services over Global Networks), hereinafter referred as Kar.

a. Regarding claims 25 and 27, Kathrow shows a method for managing components in an information technology (IT) system, comprising: receiving a first event message for a first occurrence in the IT system, the first occurrence relating to a first component (Fig. 2, item 252; column 4, line 7-18); if the first component matches at least one low-level element of a fingerprint of a model of a known component, adding the first component to an accumulator (Fig. 4: dash line; column 5, line 45-53: REPEAT; column 11, line 14-21: REPEAT). Kathrow also shows (column 4, 1st paragraph; column 11, line 62-64) a single file could be divided into multiple files. Kathrow does not show if all of the low-level elements of the fingerprint have been matched by the first component and other components, generating a command to detect further information; receiving, in response to the command, a second event message providing further details about one of the components; and using a subfingerprint of a known refined component and the further details about one of the components to discover a refined component; and further comprising: generating a

component detected message upon the discovery of the refined component; retrieving a list of elements to track for the refined component; and using the list of elements to track changes to the refined component.

- b. Kar shows (page 5; left column, last paragraph, and right column, last paragraph-page 6, left column, 1st paragraph) as a result of static analysis, the resource dependency graph could be constructed in an analogous art for the purpose of determining the resource dependency in the application service management.
- c. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Kathrow's file comparison based upon the fingerprint method to include a further depending file (refined components) comparison through an application control (Fig. 2, item 252; column 4, line 16-18) via the result of fingerprint comparison (Fig. 2, item 254; Fig. 4, items 432, 434 and 440; column 11, line 59-64) to ensure all the dependent files as suggested (column 4, 1st paragraph; column 11, line 62-64) will be compared based upon fingerprints; and to store a list of dependent files as per Kar's dependency graph suggested to further ensure all dependent files would be finger print checked.
- d. The modification would have been obvious because one of ordinary skill in the art would have been motivated to fingerprint check not only one file out of multiple files that were from the division out of a single file, but all multiple files to ensure the integrity of a file (all divided files or associated resources) in the management of applications per Kar (page 2, left column, 2nd paragraph- page 3, left column, 1st paragraph: using mid level manager to monitor the health and status of a application

service) and Kathrow (column 1, lines 36-52: using window registry to manage computer programs running under window operating system) 's teachings.

- e. Regarding claim 26, Kathrow shows the method of claim 25, wherein the first occurrence is one of a file creation, file deletion, file modification, registry key creation, registry key modification, and registry key deletion (column 4, line 60-65).

Together Kathrow and Kar disclosed all limitations of claims 25-27. Claims 25-27 are rejected under 35 U.S.C. 103(a).

10. Claims 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kathrow, et al. (US 6393438 B1), hereinafter referred as Kathrow, in view of Kar et al., (An Architecture for Managing Application Services over Global Networks), hereinafter referred as Kar.

- a. Regarding claim 28, Kathrow shows a method for discovery of a refined component in an information technology (IT) system, comprising: using a fingerprint of a model of a known component to discover an existing component in the IT system by matching passive elements in the fingerprint with event information of the IT system (Fig. 4, item 432 and 404; column 11, line 49-62). Kathrow also shows (column 4, 1st paragraph; column 11, line 62-64) a single file could be divided into multiple files and signal (event message) input/output generation via application (Fig. 2, item 252 and 254; column 4, line 7-18). Kathrow does not show generating and transmitting a command message defined by active elements of the fingerprint to discover the refined component; receiving event information relating to the active elements of the fingerprint of the known component; and using a subfingerprint of the refined component to discover the refined component, the refined component relating to the

known component, wherein the subfingerprint of the refined component becomes active upon the discovery of the existing component using the fingerprint.

- b. Kar shows (page 5, left column, last paragraph, and right column, last paragraph-page 6, left column, 1st paragraph) as a result of static analysis, the resource dependency graph could be constructed in an analogous art for the purpose of determining the resource dependency in the application service management.
- c. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Kathrow's file comparison based upon the fingerprint method to include a further depending file (refined components) comparison through an application control (Fig. 2, item 252 and 254; column 4, line 7-18) via the result of fingerprint comparison (Fig. 2, item 254; Fig. 4, items 432, 434 and 440; column 11, line 59-64) to ensure all the dependent files as suggested (column 4, 1st paragraph; column 11, line 62-64) will be compared based upon fingerprints.
- d. The modification would have been obvious because one of ordinary skill in the art would have been motivated to fingerprint check not only one file out of multiple files that were from the division out of a single file, but all multiple files to ensure the integrity of a file (all divided files or associated resources) in the management of applications per Kar (page 2, left column, 2nd paragraph- page 3, left column, 1st paragraph: using mid level manager to monitor the health and status of a application service) and Kathrow (column 1, lines 36-52: using window registry to manage computer programs running under window operating system) 's teachings.

e. Regarding claim 29, Kathrow shows the method of claim 28, wherein receiving event information relating to active elements includes receiving an event message (Fig. 2, item 252 and 254; column 4, line 7-18: signal).

Together Kathrow and Kar disclosed all limitations of claims 28-29. Claims 28-29 are rejected under 35 U.S.C. 103(a).

Response to Arguments

11. Applicant's arguments filed on 12/04/2006 have been fully considered, but they are not persuasive.

a. In response to applicant's allegation that Keller does teach or suggest receiving real-time message, monitoring the real-time usage of resources by components and indicating that a dependency between the two components has been detected. Examiner has reviewed the rejections of claims 30 and 40, and item c of Response to Arguments section in the office action dated 05/30/2006. Examiner has re-reviewed the abstract and relevant paragraphs of Keller. Examiner believes that Keller does have all the limitations of claimed invention per claims 30 and 40. Examiner has updated claim 30 rejection to further recites additional passages from relevant paragraphs of Keller. Examiner believes that item c of Response to Arguments section in the office action dated 05/30/2006 is still application to applicant's current argument. As the title of Keller points out it is about dynamic dependencies in Application Service Management, it must be done in real-time or say runtime. This is particularly pointed out on page 4, 1st paragraph of Keller as updated in the above rejection of claim 30. The same paragraph also talks about the runtime behavior of a web server, i.e. the usage of message per applicant's specification. Here the determination of usage per Keller on DNP and web server are very much internet message based service and Keller did mention receiving and processing event according to dependency that also suggests Keller's art is built upon messaging.

b. In response to application's allegation that Fig. 4, item 440 per Kathrow does not relate to determining the existence of a component if the elements of a fingerprint match. Examiner has reviewed the cited Kathrow and its references. Kathrow is on identifying the existence of difference between two files by using fingerprint, including the usage of multiple fingerprints. Examiner here has used Kathrow to show the art of using fingerprint to identify files, i.e. component as described in applicant's specification. As Kathrow applies the fingerprint technique in the window registry which is a fundamental part of a computer system and the natural of Kathrow's application is the file/software component management in the window registry, i.e. very much IT management, the combinatory of Kathrow and Kar is just an application of specific identification method in the art of dynamic dependency determination. As Kar has shown using resource identifier (page 6, right column, item 1) and software definition file format with many attributes (page 7, left column, 2nd paragraph), it is obvious to one skill in the art to use Kathrow's window based component inventory by using fingerprint technique with Kar's network resource management by identify resource and dependency dynamically as both Kathrow and Kar are in the art of application/software component management.

c. It is the Examiner's position that Applicant has not submitted claims drawn to limitations, which define the operation and apparatus of Applicant's disclosed invention in manner, which distinguishes over the prior art. As it is Applicant's right to claim as broadly as possible their invention, it is also the Examiner's right to interpret the claim language as broadly as possible. It is the Examiner's position that

the detailed functionality that allows for Applicant's invention to overcome the prior art used in the rejection, fails to differentiate in detail how these features are unique (see item a in section 5 and items a-d in section 6). Keller and Kar are specific arts in determining dynamic dependencies among network application components. Kathrow 's art is specific art on using fingerprint to identify files/software components. Kathrow's art and Kar (and Keller)'s art are distinct however related in the generic of the computer management, i.e. network management and file/component management. The prior art of O'Neill as listed in the Remark section of office action dated 05/30/2006 and here below also related software distribution and configuration updates by using fingerprint technique. It is clear that Applicant must be able to submit claim language to distinguish over the prior arts used in the above rejection sections that discloses distinctive features of Applicant's claimed invention. It is suggested that Applicant compare the original specification and claim language with the cited prior art used in the rejection section above or the Remark section below to draw an amended claim set to further the prosecution.

- d. Failure for Applicant to narrow the definition/scope of the claims and supply arguments commensurate in scope with the claims implies the Applicant's intent to broaden claimed invention. Examiner interprets the claim language in a scope parallel to the Applicant in the response. Examiner reiterates the need for the Applicant to more clearly and distinctly define the claimed invention.

Remarks

12. The following pertaining arts are discovered and not used in this office action. Office reserves the right to use these arts in later actions.

a. O'Neill (US 6832373 B2) System and method for updating and distributing information

Conclusion

13. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Refer to the enclosed PTO-892 for details.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peling A. Shaw whose telephone number is (571) 272-7968. The examiner can normally be reached on M-F 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

pas

pas

W.C. Vaughn
WILLIAM VAUGHN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100